TotalView
Release Notes

June 2002

Version 5.0.0-5

Contents

TotalView Issues

TotalView Issues

These Release Notes are for TotalView Version 5.0.0-5, June 12, 2002.

This document contains information about bugs that existed in previous versions and bugs that still exist. Other documents that contain information you may need are:

- TOTALVIEW PLATFORMS: Lists the platforms and environments in which TotalView runs.
- SPECIAL IBM CONSIDERATIONS: Describes special considerations when running TotalView on IBM platforms.
- SPECIAL LINUX CONSIDERATIONS: Describes special considerations when running TotalView on Linux platforms.
- PATCHING: Outlines the procedures for installing patches to software provided by other vendors that must be done before you can use TotalView.

You can obtain these documents in the following ways:

- PDF and HTML versions can be obtained from www.etnus.com/Support/docs.
- PDF and HTML versions are contained with the documentation tar file that you can download when you are downloading TotalView from our web site.
- A HyperHelp version of this file can be viewed after installing TotalView by selecting the Release Notes item on the Help Menu.
- HyperHelp versions of the other files are installed in the same directory as other help files.

The manuals for this release are *TotalView User's Guide*, *TotalView CLI Guide*, *TotalView Upgrading Guide*, and *TotalView Installation Guide*. Their version number is 5.0.

TotalView News

Consult the Etnus Web site at http://www.etnus.com for a discussion of the new features contained within TotalView. If you used Version 4 of TotalView, here are other changes:

- Commands That No Longer Exist
- Using the Mouse
- X Resources and Preferences

Commands That No Longer Exist

The following Release 4.0 commands have no equivalent in Release 5.0:

- Clear All Elog
- Find Interesting Relative
- Open Action Points Window
- Reexecute Last Save Window
- Run TimeScan
- Set PC to Absolute Value
- Set Process Program Group
- Suppress All Elog
- Unsuppress All Elog

For information on the relationship between TotalView Release 4 and TotalView Release 5 commands, see the TOTALVIEW UPGRADING GUIDE.

Using the Mouse

The way you use the mouse differs in Version 5. Chapter 2 of the TOTALVIEW USERS GUIDE describes usage in more detail. TotalView now follows Motif guidelines or performs a related action, as follows:

- The left mouse button selects. Double-clicking dives.
- The middle mouse button pastes text.
- The right mouse button brings up a context sensitive menu.

X Resources and Preferences

While TotalView still allows you to place directives within an X resource .Xdefaults file, the role of this file is greatly diminished. X resources that dealt with the TotalView window are ignored at Release 5. Many other resources have been deprecated because you can now specify these attributes using the File > Preferences command.

Problems Fixed

The following problems that existed in earlier releases are fixed. Problems fixed in Version 5.0.0-5 are displayed in bold.

All Versions

- C++: Symbol problems (3112)
- **CLI: CLI now opens if an xterm isn’t in the user’s path** (1640 at 5.0.0-5)
- Diving on bad address 0x00000000 crashes TotalView (3600 at 5.0.0-4)
- “END IF” is now supported in evaluations (3517)
- **Evaluation command window: scroll bars now appear when text is lengthy** (3686 at 5.0.0-5)
- File Descriptor Problem (3217)
- **MPI: TotalView did not pick up MPI application processes when MPI lib calls came from a dlopen loaded library** (3793 at 5.0.0-5)
- **Numeric host names now allowed** (3632 at 5.0.0-5)
- **PVM: Long message problem corrected** (3805 at 5.0.0-5)
- **PVM: tvdsvr processes are not shut down on PVM virtual machine nodes when TotalView exits** (3825 at 5.0.0-5)
- **STL vector names are now displayed, even when very long** (3857 at 5.0.0-5)
- **STL vector sorting no longer fails when performing a type transformation** (3838 at 5.0.0-5)

- **Truncated stack trace from core dump when process aborts in signal handler** (3614 and 3617 at 5.0.0-5)

Compaq Alpha

- Compaq Alpha: DMPI 1.95 is not supported (2963)
- Compaq Alpha: lerr while reading /usr/shlib/libexec.so (2307) and TotalView crashes and displays “Internal error in TotalView” (3114)
- Compaq Alpha: libots symbols for OpenMP are not recognized (3271 at 5.0.0-2)

HP

- **HP: 64-bit DLL debugging now works correctly** (3813 at 5.0.0-5)
- HP: Visualizer now starts up correctly (3334 at 5.0.0-4)

IBM AIX

- IBM AIX: After setting a breakpoint in a poe program, location being displayed jumps (1319 at 5.0.0-1)
- IBM AIX: Arrays no longer displayed in reversed order (3459 at 5.0.0-2)
- IBM AIX: Call tree wrote confusing messages to TotalView’s output (3363 at 5.0.0-1)
- **IBM AIX: LD_LIBRARY_PATH Needs to be Set Due to IBM Changes** (3683 at 5.0.0-5)
- **IBM AIX: malloc_and_check errors no longer occur** (3818 at 5.0.0-5)
- IBM AIX: pow() broken (3184)
- **IBM: Stack trace problems when floating point exceptions are enabled using -qflttrap** (3530 at 5.0.0-5)

Linux

- Linux: Changing UI font and font sizes from File > Preferences Dialog Box causes an internal error (3576 at 5.0.0-4)
- Linux: Dynamic libraries were not handled with static executables (3123)
- Linux: PGI compiler produces a SEGV (3257)
- Linux: PGI Compiler—Improved module and pointer support for Fortran 90 (3624, 3607, 3606, and 3599 at 5.0.0-4)
- Linux: User threads not enabled after libpthread.so is dynamically loaded (3423 at 5.0.0-2)
- Linux—Compaq Alpha: Demangler problems (3366 at 5.0.0-2)

- Linux—Lahey/Fujitsu: Fatal error loading DWARF from Lahey/Fujitsu Fortran (3447 at 5.0.0-2)
- Linux—Lahey/Fujitsu: TotalView no longer shows the wrong address for allocatable Fortran 90 arrays (3579 at 5.0.0-4)
- **Linux—Mandrake (unsupported Linux distribution): TotalView no longer crashes when started when XLOCALEDIR is not set** (3779 at 5.0.0-5)

SGI

- SGI: cord(1) Reordering change in IRIX 6.5.11m causes truncated stack traces from libc routines (2804)
- SGI: Parameters within a Fortran 90 program created a fatal error (3319 at 5.0.0-1)
- **SGI: Process Window no longer closes when an MPT MPI job completes** (3369 at 5.0.0-5)
- **SGI: shmem—TotalView no longer crashes when attaching to a shmem process on SGI mpp** (3830 at 5.0.0-5)
- SGI: signed int: 32 problem in a struct when using KCC (2984 at 5.0.0-2)
- SGI: “Stop the Job Now” dialog box does not appear after reloading symbols with MPI (3333 at 5.0.0-2)
- SGI: KCC—Group > Reload Symbols command now works correctly on KCC 3.4b/4.0d (3407 at 5.0.0-4)

Sun

- Sun: Casting derived data display problem (3456 at 5.0.0-2)
- Sun: Fortran 90 programs compiled with Forte 6 Update 1 no longer crash TotalView (3471 at 5.0.0-2)
- **Sun: License checkout problem fixed** (3780 at 5.0.0-5)
- Sun: Setting breakpoint in templates class error (3394 at 5.0.0-2)
- Sun: Solaris Workshop 6.1—Fortran 90 complex variables are now displayed correctly (3428 at 5.0.0-4)
- Sun: SunPro C++ 5.0-compiled program no longer crash TotalView (1984 at 5.0.0-2)

Explanations

C++: Symbol problems

After setting a breakpoint and hitting Go, TotalView displayed the following error message:

```
Error: Symbol table parser, reading symbol "":
stab_typetable_entry_t::parse_type_string: Unexpected colon
token: ::(0,26)=##(0,25);:__as__1
```

(3112)

CLI: CLI now opens if an xterm isn't in the user's path

Previously, if an xterm was not in the user's path, TotalView would freeze when the user tried to open the CLI from the Root or Process Windows. (1640 at 5.0.0-5)

Diving on bad address 0x00000000 crashes TotalView

Previously, when you dove on 0x00000000, TotalView crashed, displaying the following error message:

```
Internal error in TotalView.
     Terminated
```

(3600 at 5.0.0-4)

"END IF" is now supported in evaluations

TotalView now supports both ENDIF and END IF in evaluations. For example, you can now enter the following expression:

```
if (i==3) then
     $stop
end if
```

(3517)

Evaluation command window: scroll bars now appear when text is lengthy

If you typed a lot of text in an Evaluation Window, you could have difficulty seeing what you typed because the window wasn't designed to have Scroll Bars. The window has been redesigned. (3686 at 5.0.0-5)

File Descriptor Problem

On a multithreaded server application, TotalView could run out of file descriptors and hang. (3217)

MPI: TotalView did not pick up MPI application processes when MPI lib calls came from a dlopen loaded library

If the MPI library calls are in a library which is deferred-loaded/linked, TotalView did not pick up that it is an MPI program. This meant that TotalView did not automatically attach to the other processes.

If you attached to other processes using the Unattached page, the process became stuck in a select call in MPIR_Init.(3793 at 5.0.0-5)

Numeric host names now allowed

TotalView wrongly assumed that hosts names could not begin with a number. (3632 at 5.0.0-5)

PVM: Long message problem corrected

If the the PVM_EXPORT ENV variable is set and contains less than approximately 450 characters, everything is normal and TotalView works as expected. If the PVM_EXPORT ENV variable contains more than approximately 450 characters, TotalView crashes with the following error:

```
Fatal error: Overlength message -- won't fit in buffer
Terminated
```
(3805 at 5.0.0-5)

PVM: tvdsvr processes are not shut down on PVM virtual machine nodes when TotalView exits

When debugging PVM jobs, TotalView spawns a tvdsvr process on each node in the parallel virtual machine. When TotalView exits, it should terminate the running tvdsvr processes on the other nodes. If they aren’t terminated, you will get a message like the following the next time you start TotalView on a PVM task:

```
TotalView: FATAL ERROR STARTING UP: A PVM tasker is already
running on host 'alien'
```

In some cases, TotalView did not terminate all of the tvdsvr processes that it spawned. This means that the user must reset the virtual machine before again running TotalView.(3825 at 5.0.0-5)

STL vector names are now displayed, even when very long

When using the STL, TotalView could abort, displaying the following error message:

```
Fatal error: form_long result too large
```

The problem was due to huge symbol names being generated. (3857 at 5.0.0-5)

STL vector sorting no longer fails when performing a type transformation

Memory became corrupted when sorting large vectors using TotalView type transformation system. (3838 at 5.0.0-5)

Truncated stack trace from core dump when process aborts in signal handler

TotalView's behavior across platforms when debugging a core file generated from an abort in a signal handler is not consistent. On AIX and LINUX the trace is truncated and you only get the signal handler; on Sun there seems to be a problem with the PC but the trace is intact; on SGI and TRU64, behavior appears to be consistent. (3614 and 3617 at 5.0.0-5)

Compaq Alpha: DMPI 1.95 is not supported

TotalView does not support DMPI 1.95. You must upgrade to DMPI 1.96. (2963)

Compaq Alpha: Ierr while reading /usr/shlib/libexec.so (2307) and TotalView crashes and displays "Internal error in TotalView" (3114)

TotalView was dereferencing a NULL pointer.

Compaq Alpha: libots symbols for OpenMP are not recognized

Compaq has changed the way they construct OpenMP backlink tokens. Compaq delivers this library both with compiler kits and with the Tru64 UNIX kit. An updated library is distributed with Compaq Fortran 5.4B. This library will also be distributed with future versions of the operating system. (3271 at 5.0.0-2)

HP: 64-bit DLL debugging now works correctly

Debugging 64-bit shared libraries on HP-UX was broken on some systems. The symptoms were an inability to set a breakpoints or step into shared libraries . (3813 at 5.0.0-5)

HP: Visualizer now starts up correctly

TotalView's Visualizer could not be started because it could not find shared libraries.

(3334 at 5.0.0-4)

IBM AIX: After setting a breakpoint in a poe program, location being displayed jumps

After setting a breakpoint in a poe job, TotalView jumped and displayed another location. The breakpoint was set correctly; however, TotalView was incorrectly display the source file. (1319)

IBM AIX: Arrays no longer displayed in reversed order

In arrays contained within a derived type, TotalView reversed the array dimensions in the Variable Window on AIX using xlf. (3459 at 5.0.0-2)

IBM AIX: Call tree wrote confusing messages to TotalView's output

Although the Call Tree window displayed information correctly, Xlib error messages were written to the xterm window from which you invoked TotalView. (3363)

IBM AIX: LD_LIBRARY_PATH Needs to be Set Due to IBM Changes

IBM will be building and distributing the MPI message queue DLL. We have added a directive to set LD_LIBRARY_PATH to include /usr/lpp/ppe.poe/lib. (3683 at 5.0.0-5)

IBM AIX: malloc_and_check errors no longer occur

This error was due to a problem in the way TotalView managed the large addess pace model. (3818 at 5.0.0-5)

IBM AIX: pow() broken

The pow() function in libbsd.a on AIX is broken. TotalView code was rewritten so that it does not use this function. (3184)

IBM: Stack trace problems when floating point exceptions are enabled using -qflttrap

If exceptions were enabled using the -qflttrap option in xlf on AIX, TotalView's stack trace was incorrect in that it didn't have .main. (3530 at 5.0.0-5)

Linux: Changing UI font and font sizes from File > Preferences Dialog Box causes an internal error

When you try to change the font and font size, TotalView crashes, displaying the following error message: "trying to free invalid block". You are able to perform this operation on other platforms. Temporarily, the control's that you can select that crash TotalView have been removed on Linux versions of TotalView. You can still set fonts by using X Resources or by setting variables in a .tvdrc file. (3576 at 5.0.0-4)

Linux: Dynamic libraries were not handled with static executables

Linux allows statically linked executables to use dlopen to dynamically load shared libraries. TotalView believed that only dynamically linked executables can do this, and so disabled dynamic library processing when it saw a statically linked executable. (3123)

Linux: PGI compiler produces a SEGV

TotalView mistakenly freed symbols associated with a common block. (3257)

Linux: PGI Compiler—Improved module and pointer support for Fortran 90

Information about Fortran 90 modules and pointers was not being processed correctly.

(3624, 3607, 3606, 3599 at 5.0.0-4)

Linux: User threads not enabled after libpthread.so is dynamically loaded

When a process dlopen()'s libpthread.so, TotalView did not automatically enable user threads. (3423 at 5.0.0-2)

Linux—Compaq Alpha: Demangler problems

TotalView did not correctly demangle C++ type names. (3366 at 5.0.0-2)

Linux—Lahey/Fujitsu: Fatal error loading DWARF from Lahey/Fujitsu Fortran

TotalView crashes and displays the following error message when it loads a program compiled using the Lahey/Fujitsu Fortran on x86/Linux.

```
Fatal error: location_t::get_opcode bad index
```

(3447 at 5.0.0-2)

Linux—Lahey/Fujitsu: TotalView no longer shows the wrong address for allocatable Fortran 90 arrays

(3579 at 5.0.0-4)

Linux—Mandrake (unsupported Linux distribution): TotalView no longer crashes when started when XLOCALEDIR is not set

Note that the only supported Linux distribution is Red Hat. Check our web site to see which versions are currently supported. (3779 at 5.0.0-5)

SGI: cord(1) Reordering change in IRIX 6.5.11m causes truncated stack traces from libc routines

cord(1) reordering changes in IRIX 6.5.11m prevent TotalView from properly obtaining the information it needed to locate addresses in the user's process. If the user halted TotalView when the top of the stack is in a libc routine, TotalView could not generate more than two frames of the stack backtrace; that is, the trace appears truncated. (2804)

SGI: Parameters within a Fortran 90 program created a fatal error

If you declared a parameter variable in Fortran 90 using the MIPSpro compilers, TotalView displayed the following error message:

```
    Fatal error: Can’t find abbrev table entry for debug entry. (3319)
```

SGI: Process Window no longer closes when an MPT MPI job completes

(3369 at 5.0.0-5)

SGI: shmem—TotalView no longer crashes when attaching to a shmem process on SGI mpp

(3830 at 5.0.0-5)

SGI: signed int: 32 problem in a struct when using KCC

TotalView crashed when processing a signed int contained within a struct in programs compiled using KCC. (2984 at 5.0.0-2)

SGI: "Stop the Job Now" dialog box does not appear after reloading symbols with MPI

The Process mpirun is parallel job. Do you want to stop the job now? Dialog Box does not appear after recompiling a program and reloading its symbols using MPT/MPI. (3333 at 5.0.0-2)

SGI: KCC—Group > Reload Symbols command now works correctly on KCC 3.4b/4.0d

When using the Group > Reload Symbols command on KCC 3.4b/4.0d, programs compiled on an SGI platform, TotalView crashed, displaying the following error message: "Internal Error in TotalView". (3407 at 5.0.0-4)

Sun: Casting derived data display problem

After diving on a pointer to an object from a derived class, casting the object in the Variable Window to the actual derived class previously caused TotalView to display the following type of warning message when using Sun Workshop 5.0 C++:

```
Couldn't find a type named "class <derived_class_name>"
```

(3456 at 5.0.0-2)

Sun: Fortran 90 programs compiled with Forte 6 Update 1 no longer crash TotalView

(3471 at 5.0.0-2)

Sun: License checkout problem fixed

TotalView cannot checkout any INCREMENT lines from the FLEXlm license manager daemon for certain Sun machines. This is a client-side problem licenses can be checked out from other machines when the license manager is run on the same Sun machine. (3780 at 5.0.0-5)

Sun: Setting breakpoint in templates class error

TotalView crashed with the following error after you set a breakpoint in a template class, telling it to set the breakpoint for all types:

```
Internal error in TotalView
```

You could move up and down a few lines, but if you went too far you received this message. (3394 at 5.0.0-2)

Sun: Solaris Workshop 6.1—Fortran 90 complex variables are now displayed correctly

(3428 at 5.0.0-4)

Sun: SunPro C++ 5.0-compiled program no longer crash TotalView

(1984 at 5.0.0-2)

Known Problems

The following sections list the problems that have been found.

- Problems on All Platforms
- CLI Problems
- Compaq Alpha Tru64 UNIX Problems
- HP HP-UX Problems
- IBM RS/6000 Problems
- Linux Problems
- Portland Group HPF 2.4 Problems
- SGI IRIX Problems
- Sun SPARC 5 Problems

You may find your problem (and its solution) documented on our website's FAQ, which is located at http://www.etnus.com/Support/faqs.html.

Problems on All Platforms

All platforms have the following problems. (The new item is indicated in **bold**.)

- Attaching to Portland Group HPF Jobs is Not Supported
- C++ Exceptions
- Checkpoint Dialog Box Problem (5.0.0-0)
- EGCS Problems
- Evaluation Point With a goto and a step

- Eval System: Fortran WRITE statement not supported (5.0.0-5)
- Eval System: Spaces Required and Case Sensitive (5.0.0-5)
- FLEXlm Hunting For Multiprocessor Features
- Fortran Arrays Whose Size Changes
- Function Static Variables May Be Invisible When Using KCC
- fvwm Version 1 Problems
- GUI: Accelerators Are Overriding Menu Mnemonics if F10 is Pressed (5.0.0-0)
- GUI: Page Up, Page Down, Up Arrow and Down Arrow Problems (5.0.0-0)
- GUI: Page Up and Page Down May Not Work in Some Panes (5.0.0-0)
- MPICH 1.2.0 Cannot Locate libtvmpich.so Library
- MPICH 1.2.1 Invocation Problems (5.0.0-4)
- Multithreaded Corefile Display Problem (5.0.0-0)
- Search Path Problems (5.0.0-0)
- Stop Command within Tools > Evaluate Dialog Box Can Corrupt Target Process
- Visualizer Can Have Font Problems under CDE
- Watchpoints Not Allowed on Registers (5.0.0-0)
- Xoftware and Motif Problems (5.0.0-0)

Attaching to Portland Group HPF Jobs is Not Supported

TotalView does not support attaching to Portland Group HPF jobs. If you attempt to attach to Portland Group HPF jobs, you may not see all of the processes that the job is composed of, and you may not be able to display distributed variables.

C++ Exceptions

TotalView does not have full support for C++ exceptions. Single-stepping over code that will throw an exception is problematic and often results in the process running away. To help with this situation, TotalView will detect when an exception throw is going to occur while single-stepping.

By default, TotalView brings up a dialog box to ask if you wish to stop the process. Answering No continues the process. Be aware that if you are stepping within the “try” block, your process may run away. Answering Yes stops the process upon entry into the system runtime routine that issues the throw. This is a temporary solution and full C++ exception handling may be provided in a future TotalView release.

This mechanism is available for all supported C++ compilers on the supported platforms for SGI IRIX 6.x, Power AIX, Alpha Compaq Tru64 UNIX, and SPARC SunOS 5 (Solaris 2.x) platforms.

The following user interface controls are available for turning this dialog box on and off:

- X *resource*: totalview*warnStepThrow: { true | false }. If you do not specify this X resource, the default value is true.
- *Command-line option*: -warnStepThrow={ true | false }
- *Menu command item*: You can select the Warn about C++ exceptions during single step operations checkbox within the File > Preferences dialog box's Options Page. This option lets you toggle the warning on and off.

If this option is turned off, TotalView does not catch C++ exception throws during single-step operations. This may cause the single-step operation to lose control on the process and cause it to run away.

Checkpoint Dialog Box Problem

If a checkpoint operation fails, the Process Window pops to the top, obscuring the Checkpoint dialog box. As this is a modal dialog box, TotalView appears to be hung. (3199)

EGCS Problems

The abbreviation table created EGCS 2.91.66 is incorrect. If TotalView prints an error message. This abbreviation table problem was fixed in gcc 2.95.2.

You have two alternatives:

- (Preferred) Upgrade to 2.95.2 or later
- Use the –g option instead of –ggdb3.

Eval System: Fortran Intrinsics such as WRITE and COS Not Supported

TotalView does not support Fortran intrnsics such as write, sin, cos, and tan in its evaluation system; i.e., in watchpoints, evaluation points, and in the Evaluation Window. (3842)

Eval System: Spaces Required and Case Sensitive

TotalView assumes that you will use spaces and lowercase when entering statements in its evaluation system; i.e., in watchpoints, evaluation points, and in the Evaluation Window. For example, the following fails:

```
if(i.ge.10.and. i.le.20) $stop
```

This will also fail:

```
if(i.ge.10 .and. i.le.20) $STOP
```

This succeeds:

```
if(i.ge.10 .and. i.le.20) $stop
```

(1985)

Evaluation Point With a goto and a step

If an evaluation point executes a goto statement or an assembly language transfer of control instruction, and you use the step or next command at the line where the evaluation point is enabled, TotalView continues the program and the step or next command does not complete. To regain control, type ^C into the program window.

FLEXlm Hunting For Multiprocessor Features

When FLEX*lm* reads your license.dat file, it hunts for multiprocessor feature lines when you start a debugging session with more than two processors. If the following message appears, it can be ignored:

```
(toolworks) UNSUPPORTED: "TV/<hdwr>-<OS>/MP/<n>"
```

Fortran Arrays Whose Size Changes

When a Variable Window displays a single element of a Fortran array that has runtime bounds (that is, assumed shape, assumed size, allocatable, or a pointer), and the actual bounds change, the value displayed in the Variable Window applies to the wrong element in the reshaped array.

You will only see this problem when all of the following conditions occur:

- The size of your Fortran array changes
- You use the View > Variable command
- You are only displaying a single element, either because you have dived, or because you had used an array index with a command that displays this window.

To overcome this problem, display the whole array, then dive to the element that you want to see. Alternately, if you select the specific element of interest by setting the slice expression rather than by diving, the correct element always displays, even if the array changes shape.

fvwm Version 1 Problems

There are problems with the fvwm version 1 window manager. Some users have reported that TotalView triggers bugs in version 1.22d of the fvwm window manager (and presumably earlier versions, too). However, The last release of fvwm version 1 (release 1.24r) is believed to work correctly with TotalView. Therefore, if you are using the fvwm version 1 window manager, we recommend that you ensure that you are using version 1.24r. We have not tested any later versions. You can find full details on fvwm at http://fvwm.math.uh.edu/.

Function Static Variables May Be Invisible When Using KCC

The KCC compiler moves a static variable from the function in which it is declared and places the declaration at file or global scope. It also mangles the name to show that the variable ought to be at function scope. Unfortunately, TotalView does not understand this mangling.

GUI: Accelerators Are Overriding Menu Mnemonics if F10 is Pressed

TotalView's use of single letter accelerators such a f and v override the menu mnemonics (the underlined letters) when the menubar is selected using the F10 key. (2756)

GUI: Page Up, Page Down, Up Arrow and Down Arrow Problems

These keys do not work if you are in the scroll bar area. (3081)

GUI: Page Up and Page Down May Not Work in Some Panes

Page Up and Page Down do not work in some cases after clicking in a pane. The arrow keys, however, continue to work. This is most often seen in CDE 1.4 running on Solaris. (3082)

MPICH 1.2.0 Cannot Locate libtvmpich.so Library

If you are running MPICH 1.2.0, TotalView cannot find the libtvmpich.so library. Installing patch 4959 (downloadable at http://www.mcs.anl.gov/mpi/mpich/buglist-tbl.html) fixes this problem.

MPICH 1.2.1 Invocation Problems

In MPICH version 1.2.1, the mpirun_dbg.totalview shell script starts TotalView with the correct options. As distributed with MPICH 1.2.1 (which is contained in util/mpirun_dbg.totalview.in), the mpirun_dbg.totalview script has two serious problems:

- A $ is missing from the cmdLineArgs parameter:

  ```
  totalview $prognamemain -a cmdLineArgs -p4pg $p4pgfile\
            -p4wd $p4workdir -mpichtv
  ```

  That is, cmdLineArgs should be $cmdLineArgs. As distributed, this script trashes command line arguments.
- The script does not use the TOTALVIEW environment variable. (This was used by previous versions.) This is very especially important when invoking the CLI using the totalviewcli command.

A patch is available for these problems in the patch.all file from mpich. There's also a separate patch file #5776 that just fixes just the TOTALVIEW problem; that is, it doesn't fix the $cmdLineArgs problem.

Unfortunately, the patch in patch.all also adds support for the -t option to mpirun, but does so incorrectly. The support for -t is not correct anywhere.

You can fix this problem by editing the end of util/mpirun_dbg.totalview.in (and the created file mpirun_dbg.totalview file) to read as follows:

```
#
if [ $just_testing = 1 ] ; then
    doitall="echo"
else
    # We need the eval to handle arguments containing blanks.
    doitall="eval"
fi
# Note that this is run from within the mpirun.ch_p4 script
# only (!), so the argument list is p4 specific.  FIX ME!
#
$doitall $tvcommand $prognamemain -a $cmdLineArgs \
          -p4pg $p4pgfile -p4wd $p4workdir -mpichtv
```

The problem is that mpirun_dbg.totalview is run as a subshell that does not use the "." command. This means that only variables available to the subshell are the ones exported to it. This does not include just_testing. It is not desirable to export this variable since that will pollute the user's environment. The only way to pass this information to the subshell is to pass a -testing argument to the mpirun_dbg.totalview script when the just_testing variable is 1.

The simplest way to patch this up would be to comment out the lines:

```
if [ $just_testing = 1 ] ; then
     doitall="echo"
else
     # We need the eval to handle arguments containing blanks.
     doitall="eval"
fi
```

and replace $doitall with eval. This will produce a working script that does not support the -t switch. A better fix would support -t, but this requires more extensive modification.

Multithreaded Corefile Display Problem

TotalView shows the wrong initial thread in a multithreaded corefile. It should display the file that received the signal. (3267)

Search Path Problems

After you have set the search path, only Process Windows for the same process as the Process Window from which you selected the Search Path dialog are updated. Independent Process Windows are not updated. If you choose the File > Search Path command form the Root Window, no Program Windows is updated. (3243)

TotalView does not display a "." when you initially bring up the File > Search Path dialog box even though the current directory is in your path. (3265)

Stop Command within Tools > Evaluate Dialog Box Can Corrupt Target Process

The use of the Stop button within the Tools > Evaluate window may corrupt the target process. When the following three conditions hold:

- You call a function from the expression window.
- The process stops at a breakpoint inside that function call.

- You select the Stop command in the expression window to abort the execution of the expression,

TotalView shows the thread in an inconsistent state: the target threads are still at the breakpoint inside the function, but the stack backtrace shows it where the expression was invoked. As a result, TotalView may: (a) correctly show the source line where the process really is (from whatever line you invoked the expression); or (b) it may mistakenly show the line of the breakpoint in the function.

Further, if you try to continue the target process, one of the following will happen:

- On Alpha, RS/6000, and IRIX6-MIPS, the target process hangs.
- On Sun 4 and Sun 5, the target process continues.

To avoid a crash or a hang, toggle the breakpoint (disable then reenable the breakpoint) TotalView is reporting as current before continuing the process. But, on Sun 5 (and on all other platforms after you've toggled the breakpoint appropriately), if the process was sitting at a breakpoint when you called the function from the expression window, TotalView immediately hits that breakpoint again.

Visualizer Can Have Font Problems under CDE

In some cases, you will receive a font-based error message when starting the Visualizer. For example:

```
Warning: Cannot convert string "-dt-interface system-medium-
                r-normal-xl*-*-*-*-*-*-*-*-*-*" to type FontSet
Warning: Unable to load any usable fontset
Warning:
    Name: FONTLIST_DEFAULT_TAG_STRING
    Conversion failed.  Cannot load font.
```

You can fix this problem in several ways:

- Add one of the following X resource definition to your .Xdefaults file:

  Visualize*FontList: fixed

  or

  Visualizer*FontList: -adobe-helvetica-medium-r-normal-*-12-*-*-*-*-*-*-*

Log out and then back in or use the xrdb program to reload the .Xdefaults file.

- Login as su to root and edit /opt/totalview/lib/Visualize to add one of these FontList definitions.
- (preferred solution) Modify the font path for your X server to include the font.aliases file from your CDE installation. On Compaq Tru64 UNIX, the font alias file can be found in /usr/dt/config/xfonts/C.

Watchpoints Not Allowed on Registers

TotalView does not support setting watchpoints on registers. If you set a watchpoint on a register variable TotalView does not display an error message. When the value in the register changes, TotalView does display an error message, but this message does not indicate what the real problem is. (3075)

Xoftware and Motif Problems

If you set Xoftware version 8 to emulate setting Motif properties, modal dialog boxes for TotalView can become system modal; that is, they prevent all other window input to any window until the window is dispatched. If some other problem occurs at this time, you will need to reboot your NT workstation.

You can avoid this problem by selecting the Windows Option tab from within the Options>Configuration dialog box and set Motif Properties to off.

Motif text acclerators sometimes do not work. (2840)

Typing an Escape while navigating in the menus may crash TotalView if your focus policy is “X”. (2714)

When displaying Help, Xoftware often mangles how it displays pictures.

CLI Problems

The following problems exist within the CLI. (This section has not changed at Version 5.0.0-5.)

- In some cases, the CLI does not let you specify a variable's scope. For example, #foo#bar#x should identify a variable x within executable bar contained within the executable named foo. You may get errors indicating that the variable is not found.
- If xterm is not your PATH and you ask for a command line window, TotalView generates no diagnostics and freezes. (1640)

Compaq Alpha Tru64 UNIX Problems

The following are known problems with this platform. (This section has not changed at Version 5.0.0-5.)

- Anonymous unions Using GNU
- Compiling with -C to Detect Subscripts
- Planting Too Many Action Points Causes Problems
- Setting a Breakpoint In a Large Shared-Memory Target Causes a SEGV
- Thread Debugging Problems On All Versions of Compaq Tru64 UNIX
- TotalView Segmentation Fault at Start-up on a Node Within a Cluster After Executing for Several Days
- Using include and #include

Anonymous unions Using GNU

The GNU compiler does not output debugging information for members of anonymous unions that are enclosed in other aggregates when using the ECOFF format on the Compaq Alpha. As a result, if you are debugging in such an environment, you will not see these kind of members while looking at a data structure that contains them. Furthermore, the debugging information for the offsets of aggregate members that follow the anonymous union is output incorrectly, so these members are displayed with incorrect values.

Compiling with -C to Detect Subscripts

Compiling with -C to detect subscript out of range errors at run-time may cause TotalView to jump to a dimension statement while stepping.

On occasion, you will see a box appearing around the line number of a dimension statement when you are stepping through a routine. Due to a compiler bug in Compaq Fortran, several assembler instructions generated for certain bounds checks are associated with the line number of the array dimension statement rather than with the line number where the subscript is checked. When you single step a bounds-checked line and if that line contains the lowest instruction addresses associated with a dimension statement, TotalView will stop at the dimension statement. TotalView steps over other bounds-checked lines properly.

Planting Too Many Action Points Causes Problems

On a V4.0 or later Compaq UNIX system, using one or more TotalView commands that plant a lot of breakpoints results in an error message being displayed when you run, continue, step, or otherwise cause your program to continue or start execution.

Compaq is aware of the problem, but a fix is not yet available.

You can temporarily workaround this problem by using dbx to increase the vt_maxentries variable to something like 20,000. For example:

```
dbx -k /vmunix
assign vm_tune.vt_mapentries=20000
quit
```

You can also alter vt_mapentries using the sysconfigdb program. Consult the man page for more information.

Setting a Breakpoint In a Large Shared-Memory Target Causes a SEGV

If setting a breakpoint causes the operating system to allocate shared page tables, reading information from these pages can lead to the program getting a SEGV and TotalView exiting with a resources lost message. You can avoid this problem by setting the value of ssm-threshold to 0. For example:

```
#sysconfig -r ipc ssm-threshold=0
ssm-threshold: reconfigured
#sysconfig -q ipc ssm-threshold
ipc:
ssm-threshold = 0
```

Setting this value to 0 can degrade performance.

This problem has been reported, but a fix is not yet available.

Thread Debugging Problems On All Versions of Compaq Tru64 UNIX

Because of a bug in the Alpha thread debugging support on Compaq Tru64 UNIX, the low-level thread hold operation can allow a held thread to run. TotalView uses the low-level thread hold operation to prevent a thread from running when single-stepping another thread.

For example, assume your program has two threads, thread A and thread B. Assume that thread A is stopped at a breakpoint, and thread B is stopped elsewhere but not at a breakpoint. To continue the process (that is, both threads), TotalView must step thread A off the breakpoint. To do this, TotalView holds thread B. Then it unplants the breakpoint where thread A is stopped, sets a temporary breakpoint at the next instruction, and continues the process. Because of the hold thread bug, both thread A and thread B may run even though thread B is held. This means that thread B may miss the real breakpoint and hit the temporary breakpoint instead.

The following behaviors can indicate the presence of this bug:

- Threads miss breakpoints.
- Threads do not evaluate interpreted breakpoints.
- You see undeserved process stops, that is, the process may stop unexpectedly after evaluating an interpreted breakpoint.
- During single-stepping operations, threads other than the thread being stepped run.

TotalView Segmentation Fault at Start-up on a Node Within a Cluster After Executing for Several Days

TotalView gets a segmentation fault at start-up on a node within a cluster after executing for several days. This problem occurs over time on new nodes until no node is capable of running TotalView.

Compaq has determined that this is an NFS problem. You can fix this problem by applying the patch found at the following location:

http://ftp1.support.compaq.com/public/osf/v5.0a/

(pr 2153)

Using include and #include

If you compile Fortran 90 files with include or #include statements on the Compaq Alpha platform using the Compaq Fortran V5.0 compiler (or earlier), TotalView may show line numbers following the include statement at incorrect lines. This problem is fixed by the Fortran V5.1 compiler.

HP HP-UX Problems

The following are known problems with this platform. (This section has not changed at Version 5.0.0-5.)

- Backtrace Problems While Stopped in Some Stubs in 32-Bit Applications
- Backtrace Problems in 64-bit Application (5.0.0-0)
- Debugging shared libraries (5.0.0-0)
- NFS filesystems Must Be Hard-Mounted
- Problem with Cray Pointers
- Single-Stepping

Backtrace Problems While Stopped in Some Stubs in 32-Bit Applications

Some versions of the ld linker generate incomplete unwind information for relocation and export stubs. If TotalView is stopped in one of these stubs, the backtrace display can be affected. If you are using HP-UX 11.0, you can partially solve this problem by installing the PHSS_19866 patch, which is available from HP. This patch does not fix the problem in executables or shared libraries already linked with the faulty linker and which were supplied with the system and its compilers.

Backtrace Problems in 64-bit Application

If you are debugging a 64-bit core file, the backtrace does not include the source. All that is present is the strcpy() function where the program received the SEGV. (3260)

Debugging shared libraries

The dynamic library loader on HP-UX loads shared libraries into shared memory. TotalView does not write breakpoints into shared memory because these breakpoints could cause unrelated programs to fail.

If you need to single-step or set breakpoints in shared libraries, you must set your application to load those libraries in private memory. This is done using HP's pxdb command.

```
pxdb -s on  appname  (load shared libraries into private memory)
pxdb -s off appname  (load shared libraries into shared memory)
```

For 64-bit platforms, use pxdb64 instead of pxdb. At the present time, the version of pxdb64.exe supplied with HP's compilers does not work correctly and you must install patch PHSS_20122. You can download it from:

ftp://us-ffs.external.hp.com/hu-ux_patches/s700_800/11.X/PHSS_20122

With some versions of the linker after 11.17, using pxdb64 no longer works. This problem has been reported to HP. Contact Etnus support for information about working around the pxdb64 problem. (3218)

NFS filesystems Must Be Hard-Mounted

The debugger interface provided by HP-UX requires that the executable being debugged cannot be on a soft-mounted NFS filesystem.

Problem with Cray Pointers

The HP compiler does not emit a symbol for the “pointee” information for a Cray pointer. For example:

```
pointer (iptr, ixx())
iptr = malloc(100)
```

Because no information exists for symbol ixx, you are not able to look at it. You can, however, modify the pointer's type and then look at its contents. For example, you could change the type of iptr from integer to a a pointer such as <real*4>*, dive through it, then add bounds to the type.

Single-Stepping

TotalView does not allow you to step into functions that have not been bound by the run-time linker. If you wish to step into such a function, set a breakpoint at that function and run to the breakpoint.

You can avoid this problem by linking the executable being debugged with the -B immediate option or by calling chatr with the -B immediate option.

IBM RS/6000 Problems

The RS/6000 platform has the following known problems. (New problem indicated in **bold**.)

- AIX May Only Create a Partial Core File
- Breakpoints Can’t Be Set in Some Shared Libraries (5.0.0-0)
- Calling Dynamic Objects From Evaluation Window
- Cannot Start TotalView on MPI Tasks After Installing PTF Set 4 on AIX 4.3.3
- GNU Demangling Problem
- GPFS File System Not Supported
- Multithreaded Problems
- Parallel Session Running in the Background Can Hang TotalView
- poe Interferes With a Standalone CLI’s use of stdin
- ptrace Attaching Fails
- Process Contention Scope Not Supported
- pthdb_pthread() Returns an Empty pthread List
- **Signals Not Delivered to the Thread the User Requested**
- XL Fortran Problems Generating Incorrect Section Numbers

AIX May Only Create a Partial Core File

Recent versions of AIX (4.1 or later) dump a partial core file by default. In general, a partial core dump contains only enough information to give a stack backtrace for the faulting thread. User data sections as well as some other potentially useful information are only available in a full core dump.

To force a full core dump on AIX, you must set a signal flag with sigaction for the signal that caused the core dump. For example:

```
struct sigaction act;
act.sa_handler = SIG_DFL;
if (bigcore)
    act.sa_flags = SA_FULLDUMP;
else if (smallcore)
    act.sa_flags = SA_PARTDUMP;
sigaction (SIGSEGV, &act, 0);
```

Breakpoints Can't Be Set in Some Shared Libraries

Under some circumstances, TotalView may not be able to set a breakpoint in a shared library. We have reported this problem to IBM. (2635)

Calling Dynamic Objects From Evaluation Window

If a routine in a dynamic object is called from the expression window, and if the target routine is never called from the main program, TotalView refuses to call the routine.

Cannot Start TotalView on MPI Tasks After Installing PTF Set 4 on AIX 4.3.3

After installing PTF Set 4 on some versions AIX.4.3.3, TotalView will begin executing and hang after you answer **Do you wish to stop the parallel tasks** question. If you have this problem, please send email to support@etnus.com.

GNU Demangling Problem

Some small C++ programs compiled with the GNU compiler on AIX may not be recognized by TotalView as having been compiled with the GNU compiler. In these cases, TotalView will not demangle various program names. To make TotalView demangle the names in these programs properly, specify **-demangler=gnu** on the command line.

GPFS File System Not Supported

Version 5.0.0-5 will not work with versions of GPF earlier than version 1.3 due to limitations in that file system. No file that must be read by TotalView should be stored on a pre-version 1.3 GPFS system.

If you attempt to debug a program that has executable components (the image or dynamic libraries) on GPFS, you will see error messages like this from TotalView:

```
aix_lookup_symbol_in_load_module: Failed to map module; errno = 109
```

Multithreaded Problems

You may experience some problems when debugging multithreaded programs, because of limitations in the **ptrace()** operating system call.

The following problems can show up while you are debugging multithreaded applications:

1. When a thread stops (for example, hits a breakpoint) all the other threads stop. If any of the other threads stops while in a system call (for example, read(), sleep(), select(), etc.), however, ptrace() does not allow the debugger to read the thread's registers. As a result, TotalView:

   - Cannot display the registers, including the program counter; but does display the stack pointer
   - Cannot show you which system call is being executed
   - Cannot single-step using the *step* or *next* command, but *out* and *run to* work
   - Cannot display the top stack frame

   If you have a multithreaded application that makes a lot of system calls, it is possible that most of your threads are not fully debuggable whenever one of them stops.

2. TotalView shows you which threads are stuck in the kernel by displaying their state as In Kernel (K).
3. When a thread is created or destroyed, the system does not notify the debugger of this event. As a result, the list of threads displayed by TotalView may be stale when the program is running.
4. If the process stops for any reason, TotalView automatically updates the thread list. You may also type View > Reset to force the thread list to update.

Parallel Session Running in the Background Can Hang TotalView

On AIX systems, TotalView can hang if you have a TotalView parallel debug session running in the background in an xterm window, and you type anything in the underlying xterm window while the poe process is stopped. Type the fg command in the TotalView xterm window to clear up this condition.

poe Interferes With a Standalone CLI's use of stdin

Because poe tries to manage stdin on behalf of its target processes, a CLI invoked directly from the shell cannot read from stdin. If your target processes do not use stdin, using the -stdinmode none option to the poe command allows the CLI to use stdin. Unfortunately, this option is incompatible with the poe command's -cmdfile option.

If your processes do use stdin, your only recourse is to redirect stdin from within the CLI. For example:

```
drun < in.txt
```

(pr 2078, 2422)

ptrace Attaching Fails

Versions of the AIX kernel after AIX 4.3.3.1 contain a bug that causes a ptrace() attach to fail for some programs. In particular, attaching to a Parallel Environment program may fail. You can solve this problem by installing IY10784, whose description is:

> IY10784: ATTACH FAILS TO THE CHILD PROCESS OF A ROOT PROCESS

Process Contention Scope Not Supported

On AIX 4.3.1, 4.3.2, and 4.3.3 systems, TotalView supports debugging pthread programs running in pthread-compatibility mode or pthreads scheduled in system contention scope, that is, each pthread is bound to a kernel thread (the 1:1 thread scheduling model). TotalView does not support process contention scope, that is, multiple pthreads scheduled in user mode (M:N thread scheduling model).

On AIX 4.3.1, 4.3.2, and 4.3.3, when using TotalView to debug a program built with libpthreads.a, you must force the 1:1 model using the procedure outlined in Forcing 1:1 Thread Scheduling Mode on RS/6000 Systems, which is contained in the IBM SPECIAL CONSIDERATIONS document.

pthdb_pthread() Returns an Empty pthread List

Sometimes when a process is stopped and the pthdb_pthread() function is used to obtain a list of pthreads, the returned list is empty even when there are pthreads. (TotalView displays a console message saying that there are no more threads.) You can fix this problem by applying the APAR IY06378 patch to your system. The procedure for obtaining and applying patches is described in RS/6000 System Patch Procedures, which is found in the IBM SPECIAL CONSIDERATIONS document.

Signals Not Delivered to the Thread the User Requested

If you ask TotalView to continue a thread with a signal, TotalView can only send it to the thread that caused the last exception. TotalView cannot send it to any thread that you choose. In most cases, this means that the thread you ask to have signaled will not be the one that actually receives the signal. This problem does not affect reforwarding of signals that were originally received by a process; for example, signals for a SEGV handler.

XL Fortran Problems Generating Incorrect Section Numbers

Code compiled with the XL Fortran for AIX compiler Versions 4.1 and 5.1 may contain incorrect section numbers for bstat (.bs) and estat (.es) symbols. TotalView detects any incorrect section numbers and generates a warning in a dialog box for the first such problem only. TotalView notes any additional incorrect section numbers on its message output only. Symptom: common blocks have invalid addresses.

Patches for the 5.1 compilers are available through the normal AIX FixDist WEB. See the Patching document for downloading instructions, then download PTF U457231, which is for xlfcmp.5.1.0.2. You will be downloading all of the following filesets:

Filesets needed for selected item	Information file	Byte size
xlfcmp.5.1.0.2	README	18824192
xlfrte.5.1.0.2	README	24093696
xlsmp.rte.1.0.0.1	README	63488

Linux Problems

The following problems exist on all Linux Platforms. (This section has not changed at Version 5.0.0-5.)

- exec() Not Yet Supported
- GCC g77 Problem with Common Blocks
- KDE Problems
- LIBDBFORK Problems
- Stepping into System Routine Backtrace Problem (5.0.0-0)

- Thread Debugging and errno

exec() Not Yet Supported

Debugging threaded programs (pthreads) that call exec() is not yet supported.

GCC g77 Problem with Common Blocks

The GCC g77 compilers do not output debugging information for common blocks. Consequently, TotalView cannot show the values of variables in common blocks.

KDE Problems

Some users have reported that TotalView menus appear without a surrounding box when using KDE. The only workaround for this problem is to use a different window manager.

At other times, the entire display goes completely blank after right clicking in the source pane of tvmain after the process window appears. This problem is intermittent and we have not been able to reproduce it so that it can be analyzed.

LIBDBFORK Problems

The Linux implementation of the ptrace() debug function is flawed. This implementation reassigns the parent process of the process being debugged to TotalView. This means that an error will occur when the process's real parent attempts to *wait* for it because it will not find the child (as the child now belongs to the TotalView).

As this is a Linux kernel problem, TotalView cannot be patched to solve this problem. (The same problem will occur with any debugger on Linux). (2100)

Stepping into System Routine Backtrace Problem

If you step into a system routine such as printf(), TotalView may stop in the dynamic linker before its information has been fixed up. In this case, you may not be able to return out of this place.

You can avoid this problem by setting LD_BIND_NOW to 1 (see Appendix B of the TOTALVIEW USERS GUIDE for more information) or you can use the Next command instead of the Step command. (3015)

Thread Debugging and errno

When using pthreads on Linux, the errno variable is actually a C macro defined as follows in bits/errno.h:

```
#define errno (*__errno_location ())
```

This definition allows each thread to have its own errno value. Unfortunately, the program does not contain information that allows TotalView to find this thread specific errno value and there remains a single global errno variable still exists.

The result is that displaying errno in any thread other than the initial one in a process is likely to be very misleading, since you will see the global errno variable, rather than the per-thread value accessed by your code through the above macro.

For platform-specific information, see:

- Linux x86 Problems
- Linux Alpha Problems

Linux x86 Problems

The following problems occurs on Linux X86 Platforms. (This section has not changed at Version 5.0.0-5.)

- KAI Guidec Aborts If a Thread is Awoken With an EINTR
- PGI Problems
- Watchpoint Triggering Causes a SEGV

KAI Guidec Aborts If a Thread is Awoken With an EINTR

When running 3.8 KAI Guidec, your program calls abort if a thread is awoken with an EINTR from the system call in which it was waiting. The unmodified libraries abort with the message error: __kmp_launch_monitor: cond_timedwait: Interrupted system call (4). You can fix this problem by contacting KAI (http://www.kai.com) and requesting patch GUIDE 3.8 k320406.

PGI Problems

Several problems exist when debugging PGI F90 programs.

- PGI F90 compilers do not currently emit debug information for module variables. Consequently, all modules appear empty.
- pgf90 pointer variables that have negative strides are incorrectly displayed.
- Problems exist when demangling variables and module names containing "_" characters.

Watchpoint Triggering Causes a SEGV

If you are using a 2.2 series kernel (at least up through 2.2.14), triggering a watchpoint causes the process to execute a SEGV or an illegal instruction.

This is a kernel bug—the kernel is not correctly saving the value of the debug status register. TotalView uses this register to determine that a data breakpoint has occurred.

Older 2.0 series kernels do not have this bug.

You can correct this problem by altering and recompiling the 2.2.13 kernel. Here is the diff showing the change you must make:

```
--- traps.c.orig      Thu Dec  9 21:39:40 1999
+++ traps.c     Thu Dec  9 21:49:13 1999
@@ -354,10 +354,11 @@
    unsigned int condition;
    struct task_struct *tsk = current;

+      __asm__ __volatile__("movl %%db6,%0" : "=r" (condition));
+      tsk->tss.debugreg[6] = condition;
+
if (regs->eflags & VM_MASK)
goto debug_vm86;
-
-      __asm__ __volatile__("movl %%db6,%0" : "=r" (condition));

/* Mask out spurious TF errors due to lazy TF clearing */
if (condition & DR_STEP) {
```

Linux Alpha Problems

The following problems occur on Linux Alpha Platforms. (This section has not changed at Version 5.0.0-5.)

- Occasionally, an MPICH program being debugged by TotalView may hang on startup.
- The Compaq Fortran compiler generates incorrect debugging information for some leaf routines. This leads to missing backtraces. However, if you step to the return statement, you will see the backtrace.
- If code compiled with the Compaq compilers (ccc and fort) is linked into the same executable image as code compiled by compilers from the GNU Compiler Collection 2.95.2 (gcc 2.95.2) or later, incomplete or misleading stack backtraces can occur. This problem occurs if object files from differing compilers are mixed within the same executable file. (Using dynamic libraries produced by one compiler with an executable produced by another should work.)

Portland Group HPF 2.4 Problems

The Portland Group HPF has the following problem. (This section has not changed at Version 5.0.0-5.)

- The Portland Group HPF compiler generates bad debugging information for TotalView in cases where the compiler needs to generate static initialization subprograms. The symptom of this problem is that some line numbers in the HPF source window are not associated with actual Fortran source and TotalView either disallows setting breakpoints at some lines or it sets the breakpoint in the wrong place. This bug occurs quite often with 'WHERE'. constructs.
- The Portland Group HPF compiler generates incorrect debug information related to type definitions. The error message that you will see is:

  ```
  Fatal error: extract_diehdr, previous die not completely consumed
  ```

  (3306 at 5.0.0-1)

SGI IRIX Problems

The following are known problems with this platform. (This section has not changed at Version 5.0.0-5.)

- Arrays in main Are Not Found Unless Declared in common
- Attaching To SHMEM Jobs Is Not Supported
- Bad Template Names May Be Generated
- Cray Pointers in common Blocks Broken

- Evaluation System Cannot Access Fortran 90 Up-level Variables
- Evaluation System Forces Real Function Result into a long Temporary
- Formal Parameter Target Cannot be Shown
- Fortran 90 Pointer Variables Not Correctly Identified
- Fortran Arrays With Runtime Bounds Display Problem
- #include and -cpp Used Together in Fortran 90
- KCC Does Not Put Original File Name Into Symbol Table
- main Not Found by TotalView
- Pointers With Unlimited Bounds Shown With Bound of 1
- Single-stepping Inadvertently into System Routines
- Source Code Not Found For System Routines Complaint
- Virtual Base Classes: Showing Multiple Instances

Arrays in main Are Not Found Unless Declared in common

If an array is declared in main, the SGI MIPSpro 7.3.3 compiler does not create debugging information for the variable. Consequently, TotalView does not know that the array exists. You can workaround this problem by placing the array in a common block.

Attaching To SHMEM Jobs Is Not Supported

TotalView does not support attaching to SHMEM jobs. If you try, you may not see all of the job's processes and the process leader may not be properly identified; this could hang your job.

Bad Template Names May Be Generated

Because of a compiler bug in the SGI 7.2 and 7.2.1 compilers, bad template type names may be generated for certain template instantiations. This problem is fixed by Patch 3492: MIPSpro 7.2.1 C++ front-end rollup #4, which is available at the SGI Support Web Site.

Cray Pointers in common Blocks Broken

The debugging information generated by SGI 7.3 Fortran compiler for the targets of Cray pointers contained within common blocks contains the wrong address. Here is an example:

```
common a1(1000)
common /ptrs/ jj,iparray,kk
```

```
pointer (iparray,array)
iparray = loc(a1)
end
```

array is a real variable that is the target of the Cray pointer iparray. Because the address is wrong, TotalView cannot show you the correct values for the iparray variable. This bug has been reported to SGI. (The SGI 7.2.1 and earlier versions of the compiler do not have this bug.)

Evaluation System Cannot Access Fortran 90 Up-level Variables

Access to F90 up-level variables does not work in the evaluation system. Because of SGI F90 7.2.1 and earlier compiler bugs, access to F90 up-level variables does not work from EVAL expressions. Those variables are correctly located and displayed in data panes, however.

This problem should be fixed in the MIPSpro F90 7.3.1.2 compiler.

Evaluation System Forces Real Function Result into a long Temporary

When a program is compiled with the Fortran 90 compiler, the TotalView expression evaluation system erroneously converts real function results. The SGI Fortran 90 compiler fails to emit the return type of the function, so TotalView assumes that the return type of the function is a long. When assigned to a real variable, the return result of the function is erroneously converted from a long to a real, when in fact the function had already returned a real. Here is an example:

```
real function x_to_the_y_power(x, y)
```

TotalView expression:

```
real result
result = x_to_the_y_power(2.0, 4.0)
```

This problem, which does not occur with the Fortran 77 compiler, has been reported to SGI. (pr 2296)

Formal Parameter Target Cannot be Shown

Because of SGI 7.2.1 F90 compiler bugs, when that compiler is generating debugging information, TotalView can not show the target of a formal parameter Cray pointer. Consider the following code:

```
subroutine rex (rp)
```

```
        pointer (rp, p(8))
        p(2) = 6.
        P(5) = 3.
        write (*,*) "Should be 6,3 - ",p(2), p(5)
        return
   end
```

In this example, the compiler generates debugging information for p without any addressing information.

This problem should be fixed in the MIPSpro F90 7.3.1.2 compiler.

Fortran 90 Pointer Variables Not Correctly Identified

F90 pointer variables are not correctly identified as pointers because of incomplete debugging information generated by the compilers. TotalView displays the target data correctly, however.

This problem should be fixed in the MIPSpro F90 7.3.1.2 compiler.

Fortran Arrays With Runtime Bounds Display Problem

Some Fortran arrays with runtime bounds are displayed improperly. Because of a limitation in the debug output produced by the SGI Fortran 90 compilers, this happens for arrays which are the targets of pointers embedded in a user-defined type which has itself been arrayed. Consider the following:

```
type array_ptr
     real, dimension (:), pointer :: ap
end type array_ptr

type (array_ptr), allocatable, dimension (:) :: arrays

allocate (arrays(20))
do i = 1,20
     allocate (arrays(i)%ap(i))
end do
```

TotalView reports the bounds of the elements arrays%ap incorrectly because the compiler has generated invalid debug information for the runtime bounds.

This problem should be fixed in the MIPSpro F90 7.3.1.2 compiler.

#include and -cpp Used Together in Fortran 90

If source files contain #include statements and are compiled with the -cpp switch on a Fortran 90 program using the MIPSpro compilers, TotalView generates incorrect line numbers. To avoid this problem, use the standard Fortran include statement (without the -cpp switch).

KCC Does Not Put Original File Name Into Symbol Table

IRIX KCC code: TotalView fails to put the original file name (before preprocessing) into the symbol table. This prevents you from asking for the file by name until TotalView processes all the file's symbols.

If you use the --keep_gen_c option to the KCC compiler, you can use the following TotalView command: f *xxx*.int.c (where your original source file was *xxx*.C) to force full symbol processing of that file, after which you'll be able to do f *xxx*.C.

main Not Found by TotalView

TotalView will not find a Fortran 90 main program. TotalView will not display any source code if you do not use a PROGRAM statement within a Fortran 90 program. You can correct this problem by adding a PROGRAM statement to your main program. (pr 2099)

Pointers With Unlimited Bounds Shown With Bound of 1

Because of SGI 7.2.1 F90 compiler bugs, when that compiler is generating debugging information, TotalView shows the target of Cray pointers with unlimited bounds as having an upper bound of 1. Consider the following code:

```
subroutine test (ixx, n)
     common /sf/ iptr
     pointer (iptr, ita(*))
     ... etc ...
end
```

In this example, the compiler generates debug information for ita that indicates it has an upper bound of 1. This is incorrect because it has an unlimited upper bound.

Single-stepping Inadvertently into System Routines

The single-step commands sometimes step into system routines.

Source Code Not Found For System Routines Complaint

TotalView occasionally complains about not being able to find the source code for system routines (such as printf()).

Virtual Base Classes: Showing Multiple Instances

Because of SGI 7.1 C++ compiler bugs, when that compiler is generating debugging information, TotalView shows multiple instances of virtual base classes. Normally only one instance is correct, which is the one that is of type pointer to the base class.

Sun SPARC 5 Problems

The SPARC SunOS 5 and QSW CS-2 platforms have the following known problems. (This section has not changed at Version 5.0.0-5.)

- Apogee 4.0 compilers must be patched
- Breakpoints in thunks may cause crash

Apogee 4.0 compilers must be patched

The Apogee 4.0 compilers on Sun OS4 and Sun OS5 require a patch to bring them up to revision level 4.010. Follow the Apogee 4.0 Compiler Patch Procedures, which is found in the Patching document.

Breakpoints in thunks may cause crash

Using breakpoints in thunks may lead to unexpected results, including having the target program crash unexpectedly. A thunk is a small linkage routine that connects a subroutine call to the actual subroutines in a dynamic library. The SPARC SunOS 5 dynamic loader modifies the code in the thunks during program execution, which conflicts with TotalView's planting and unplanting of breakpoints. The first time through a thunk, the thunk branches to the dynamic loader, and the dynamic loader modifies the thunk to branch directly to the corresponding dynamic library routine. Subsequent trips through the thunk branch directly to the dynamic library routine.

Reporting Problems

If you are experiencing a problem with TotalView, please send a message to support@etnus.com to report your problem. Your message should include the following information:

- Your Name.
- Your Email address.
- Your phone number.
- Your company’s name.
- The TotalView version; for example, Version 5.0.0-5.
- The operating system version.
- The compiler vendor and the compiler’s version.
- A makefile or a copy of the command you use to compile your program.
- A description of the problem. This description should also include steps to reproduce and a a listing of All of TotalView’s messages and output.

How to Contact Us

Etnus LLC.
24 Prime Parkway
Natick, MA 01760
Internet Email: info@etnus.com
1-800-856-3766 in the United States
(+1) 508-652-7700 worldwide

Visit our web site at http://www.etnus.com/.

Notices

Etnus has prepared this document for the exclusive use of its customers, personnel, and licensees. The information in this document is subject to change without notice, and should not be construed as a commitment by Etnus. Etnus assumes no responsibility for any errors that appear in this document.